In the context of NoSQL databases, consistency is the act of keeping data across multiple nodes coordinated with each other to prevent data loss and repetition. It is important to keep the database running optimally.

The easiest option to implement the most consistency lies in the single server databases, which requires zero distribution. Therefore, the single server method is the recommended database type by eliminating the complexities of distribution options.

Inconsistencies can occur while distributing data across nodes, or by the loss of a node entirely. There are several methods for distributing data, and each has its own pitfalls. Sharding is aggregate oriented into clumps. Clump factors are physical location which place data on servers closest to the source of most requests. Even load which arrange aggregates, so they are evenly distributed across nodes. The pitfall for Sharding is if one shard fails, it renders all information from that shard lost and unrecoverable.

There is Leader-Follower replication which replicates data across multiple nodes, where one node is the leader, and all other nodes are the followers. All follower nodes are “Read-Only,” and the leader is “Read-Write” This is good for “Read=intensive” databases and is “Read Resilient.” Read Resilience is where the information is still readable after the leader node is no longer available. The pitfall for this model is inconsistency in the form of “Read-Write Conflict.” The Read-Write conflict occurs when the leader node goes down and an update is requested before another leader node can be implemented. The update missed is then a “Lost Update.” Peer-To-Peer replication model solves the Read-Write Conflict, though is inconsistent as well with “Write-Write Conflicts.” Write-Write Conflicts occur when two or more users try to push an update to the same data of a database.

Consistency is improved by implementing a combination of distribution methods. By combining Sharding with Replication there are multiple leaders, where each data item only has a single leader. A leader of one node can be the follower of other nodes, or a resolute leader/ follower node can be used. Peer-To-Peer combined with Sharding is a common strategy for Column-Family databases.

For an increase in consistency, it recommended to use a factor of three. A factor of three is when there are always a minimum of three leader nodes that have full Read/Write capability. When one or more leader nodes fail, the system can assign one of the follower nodes to increase its load by the Writeable data. This will give the system a high tolerance for failure.